IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Applicant: Haas	)	Art Unit: 2175
Serial No.: 09/658,303	)	Examiner: Mahmoudi
Filed: September 8, 2000	)	**ARC9-2000-0125-US1**
For: **SYSTEM AND METHOD FOR SCHEMA MAPPING**	)	March 24, 2005 750 B STREET, Suite 3120 San Diego, CA 92101

APPEAL BRIEF

Commissioner of Patents and Trademarks
Washington, DC 20231

Dear Sir:

This brief is submitted under 35 U.S.C. §134 and is in accordance with 37 C.F.R. Parts 1, 5, 10, 11, and 41, effective September 13, 2004 and published at 69 Fed. Reg. 155 (August 2004). This brief is further to Appellant's Notice of Appeal filed herewith.

Table of Contents

(1) Real Party in Interest

The real party in interest is IBM Corp.

(2) Related Appeals/Interferences

No other appeals or interferences exist which relate to the present application or appeal.

(3) Status of Claims

Claims 1-22 are pending, of which Claims 1-5, 7-14, and 16-22 have been finally rejected while dependent Claims 6 and 15 have been indicated as reciting allowable subject matter.

(4) Status of Amendments

No amendments are outstanding.

(5) Concise Explanation of Subject Matter in Each Independent Claim, with Page and Figure Nos.

As an initial matter, it is noted that according to the Patent Office, the concise explanations under this section are for Board convenience, and do not supersede what the claims actually state, 69 Fed. Reg. 155 (August 2004), see page 49976. Accordingly, nothing in this Section should be construed as an estoppel that limits the actual claim language.

Claim 1 sets forth a computer system that includes a general purpose computer (18, figure 1, page 6, line 9) that includes logic for mapping data arranged in a source schema (12, figure 1, page 6) into a target schema (16, id.). The logic includes receiving at least one value correspondence, with each value

correspondence representing a function for deriving a value of a target attribute from one or more values of source attributes (figure 2, page 7, line 15 *et seq.*; schematic source and target showing "input value correspondences"). The logic further includes grouping at least value correspondences into potential sets (block 26, figure 2, page 8, lines 12 and 13), and selecting candidate sets from at least some potential sets (block 28, id.). Candidate sets are grouped into covers (block 30, id.) and then covers are used to generate a query representing a source schema-to-target schema mapping (block 32, id.)

The references above are incorporated into this paragraph. Claim 9 recites a computer-implemented method for generating a mapping from a source schema to a target schema. The method includes generating a mapping based on a subset of value correspondences, with each value correspondence representing a function for deriving a value of a target attribute from one or more values of source attributes. Claim 9 also recites allowing a user, in a user interaction, to incrementally add or delete a value correspondence from the subset (e.g., the pruning at block 28, referenced above). A new mapping is generated based on the user interaction (see, e.g., figure 5 and discussion on page 10, lines 5-15). A representation of the new mapping is presented to the user such that the user can view the representation, id., and the user can add or delete a value correspondence embodied in the new mapping to generate another mapping.

The references above are incorporated into this paragraph. Claim 17 sets forth a computer program device has a computer program storage device (e.g., 20, figure 1, page 6, lines 9 and 10) that is readable by a digital processing apparatus, and a program on the program storage device and including instructions executable by the digital processing apparatus for performing method acts for generating a query representing a source schema-to-target schema mapping. The program includes computer readable code means for establishing plural value correspondences (figure 2, page 8, lines 6-11) and computer readable code means

for generating subsets of value correspondences such that (1) each subset has at most one value correspondence per target attribute, (2) for each subset requiring more than one source relation to undertake a mapping, a join path can be found between the relations, and (3) each subset includes at least every value correspondence (logic structure shown in Figures 3 and 4 and discussed on pages 9 and 10). Computer readable code means are also provided for generating a query using one of the subsets, with the query being representative of a source schema-to-target schema mapping (block 32, figure 2, id.)

(6) Grounds of Rejection to be Reviewed on Appeal

(a) Claim 9 has been rejected under 35 U.S.C. §102 as being anticipated by Ripley, USPP 2002/0023097.

(b) Claims 1-5, 7, 8, 10-14, and 16-22 have been rejected under 35 U.S.C. §103 as being unpatentable over Ripley in view of Morgenstern, USPN 5,970,490.

(7) Argument

As an initial matter, it is noted that according to the Patent Office, a new ground of rejection in an examiner's answer should be "rare", and should be levied only in response to such things as newly presented arguments by Applicant or to address a claim that the examiner previously failed to address, 69 Fed. Reg. 155 (August 2004), see, e.g., pages 49963 and 49980. Furthermore, a new ground of rejection must be approved by the Technology Center Director or designee and in any case must come accompanied with the initials of the conferees of the appeal conference, id., page 49979.

Appellant argues below three lines: (1) the declaration swearing behind Ripley has been improvidently ignored; (2) even absent the declaration, one or more portions of Ripley that have been relied upon do not find support in the provisional and, since Ripley was filed after the present application, these unsupported portions of Ripley do not constitute prior art; and (3) in any case Ripley does not teach or suggest the claims.

For the record, Appendix B contains both the previously-filed Rule 131 declaration, which had been signed by inventor Haas, and a newly-filed declaration which has been signed by the remaining two inventors as required by MPEP §715. Since the newly-filed declaration is the prior declaration verbatim except for using the plural in the introduction instead of the singular, thus requiring no new substantive consideration, and since the examiner did not previously object to the declaration for being signed by only a single inventor and the newly-filed declaration consequently is submitted herewith only to perfect the record for appeal, Appellant believes that submitting the remaining two signatures at this stage is appropriate.

The examiner refuses to permit the swear behind but ignores one of two reasons why the present invention should be accorded an earlier date than Ripley. Specifically, the Declaration alleges not just that an actual reduction to practice occurred before the provisional filing date of Ripley, but in the alternative that a prior conception occurred, coupled with diligence in filing the present application, a point fatally ignored in the Office Action. Note that evidence for the allegation of diligence is evidentiary testimony based on first hand knowledge, rendered under penalty of perjury, while written contemporaneous evidence for conception has already been made of record. Absent the introduction of actual evidence to the contrary, conception and diligence must be accepted.

The examiner has responded to the effect that the declaration contains only a general allegation of diligence. That is simply incorrect. The allegation under penalty of perjury attests to some very specific

facts related to diligence. Specifically, the inventor has testified based on first hand knowledge that (1) the invention was disclosed to IBM patent attorneys; (2) who worked in the normal course of business in diligently processing and filing patent applications with outside counsel; and (3) counsel were diligent in promoting the filing of a patent application in accordance with standard IBM patenting procedures. This is no mere general allegation, but rather a detailed, fact-intensive explanation of the procedures that were followed. Does the examiner wish to review and pass judgement on IBM's course of business in processing the thousands of disclosures it generates each year? Or does he have some reason for disbelieving the specific facts alleged in support of diligence? He gives no indication other than the above-mentioned incorrect allegation that only generalities have been alleged in support of diligence, which a cursory reading of the declaration proves is untrue.

Apart from the weight of the declaration, the claims are patentable for two further reasons. First, the latest edition of MPEP §2136.03(III) (May, 2004) now makes explicit that an examiner can accord a reference a provisional filing date only to the extent that the underlying provisional application supports the relied-upon disclosure. This case illustrates the importance of following the rules, because, taking just the sections of Ripley relied on in the anticipation rejection, paragraphs 19 and 24 do not appear in the copy of the Ripley provisional application. They were added to the disclosure when Ripley (the publication, not the provisional) was filed, after the present filing date. The same appears to be true of, e.g., paragraphs 90 and 109.

Refreshingly if somewhat surprisingly the examiner admits that the relied-upon paragraphs 19 and 24 of Ripley do not appear in the provisional. The examiner attempts, however, to discern, on page 14, lines 4-11 the substance of these paragraphs, without further explanation. Appellant will fill in the blanks.

Paragraph 19 teaches, in response to a message being returned that no match exists between a child source element and a child target element, allowing a user to define a "match". But page 14, lines 4-11 of the provisional (and referred-to figures 8-10) say nothing about a "match" at all, much less defining one, much less still in response to some sort of message. Instead, it merely teaches that a user can store and retrieve "pairings", whatever they are, and that a user can read, write, delete, and locate maps. Not exactly a ringing echo of the relied-upon paragraph 19.

Similarly, paragraph 24 is pretty hard to make out on the provisional. Nothing in the relied-upon section of the provisional mentions, for instance, determining anything about "child" elements, or separating data from a source element and applying it to a target "child", or comparing anything, much less determining a match or copying something to somewhere when a match is found.

Various other portions of Ripley have been relied on in the Office Action. Because Ripley was filed after the present application, these portions are *prima facie* not prior art. It is the examiner's burden to prove his case to the contrary. More particularly, if the examiner wishes to rely on a part of Ripley, the examiner should explain why something that on its face is not prior art actually is prior art, by identifying where in the Ripley provisional the portion is supported. It is not Appellant's burden to examine the provisional on behalf of the examiner and ferret out support for his rejections. Stated differently, it is up to the proponent of evidence to positively establish the evidence; it is not the burden of the party opponent to disprove a critical date that has not been satisfactorily established by the proponent in the first place, although Appellant has done so for some of the relied-upon portions of Ripley in an effort to promote prosecution. Having failed to carry his burden of explaining why the relied-upon portions of a document that *prima facie* is not prior art are in fact supported by the Ripley provisional, the examiner must be reversed.

(a)

Regardless of its status as prior art, Ripley does not teach or suggest the present invention. Specifically, in the anticipation rejection of Claim 9 it is alleged that paragraphs 19, 24, 80, and 81 teach the claimed "generating a mapping based on at least a subset of value correspondences, with each value correspondence representing a ***function*** for ***deriving*** a value of a target attribute from one or more values of source attributes", but this is not what Ripley teaches. All the relied-on sections of Ripley teach is using a ***mapping*** from one data structure to another. Nowhere does Ripley say anything about a value correspondence that is a ***function*** used to derive values in a target schema from those of a source.

Specifically, paragraph 19 simply mentions defining "matches"; paragraph 24 teaches allowing a user to define "datatypes" and "matchings" without relating the two to each other and without intimating that either one is a "function". Paragraphs 80 and 81 mention user-defined mappings. Thus, none of the cited sections of Ripley teach the presently claimed "value correspondence", much less that it represents a function for deriving a value of a target attribute from one or more values of source attributes. There is no "deriving" of a target attribute from a source attribute done in Ripley, much less one that uses a function.

If the examiner is reading the term "derive" to encompass a mapping under the guise of broad claim interpretation during prosecution, the examiner must bear in mind that MPEP §2111.01 permits not just any broad interpretation, but rather the broadest reasonable interpretation that one skilled in the art would give the term. This being the case, there is no evidence of record that the skilled artisan would regard mapping a database entry in one data structure directly into another data structure as taught by Ripley would be the same thing as "deriving" an entry in a target schema from a source schema (as by, e.g., the concatenation mentioned in the present specification on page 8, lines 1-5). Absent evidence that the skilled artisan regards

"derive" and "map" to be synonyms, the rejection cannot stand, In re Dembiczak, 175 F.3D 994, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999) (the range of sources available does not diminish the requirement for actual evidence, and "broad conclusory statements..., standing alone, are not evidence").

(b)

Apart from the above, it alleged in the obviousness rejection that Ripley, paragraph 72 and claim 5, groups "value correspondences" into sets, as required by, e.g., Claim 1. All paragraph 72 mentions is casting things in XML, and all claim 5 states is that data types, not the claimed value correspondences, can be nested into groups. Thus, assuming *arguendo* that the equation of the user-defined mappings in paragraphs 19, 24, 80, and 81 to the claimed "value correspondences" is correct, it is not the relied-upon mappings that claim 5 groups, but rather the data types being mapped. Accordingly, the rejection is internally inconsistent and cannot stand.

Moreover, the rejection alleges that paragraphs 80 and 81 of Ripley teach selecting candidate sets from potential sets of value correspondences as otherwise required by, e.g., Claim 1. Apart from the fact mentioned above that there is insufficient evidence of record that Ripley even teaches "value correspondences" as that term is understood in the art, paragraphs 80 and 81 simply teach that a mapping is searched for and if one is not found, "error" is returned. No mention of "candidate" anything, much less that candidate sets are selected from potential sets of value correspondences.

It is alleged that paragraphs 73-76 teach grouping candidate sets into covers, again a requirement of Claim 1. Appellant is unable to discern anything remotely approaching what is alleged is these paragraphs, and does not expect the Board to have any better luck.

It is alleged that paragraphs 23 and 84 teach using a cover to represent a mapping. For reasons above, this allegation does not appear to be based on any rational understanding of the reference being applied.

Additionally, the proferred suggestion to combine the query of Morgenstern with Ripley (to identify data to be moved from the source to the target) comes unaccompanied by the requisite *prior art* citation of support, and thus fails the standard of properly establishing a *prima facie* case of obviousness, see MPEP §2143.01. Indeed, the proferred suggestion to combine ignores the fact that Ripley does not need a query to do what the examiner proposes, since Ripley already identifies data to be moved from the source to the target without the need for a query and since the examiner admits that Ripley nowhere teaches a query for any purpose. Accordingly, the skilled artisan looking at Ripley would find no motivation to use something that is admittedly not taught in Ripley to do something that Ripley already attends to by other means, on the basis of a secondary query reference that has nothing to do with mapping a source schema to a target schema. For this additional reason, the rejection cannot stand.

The allegations regarding various dependent claim limitations likewise cannot stand, because the allegations all rely on reading things into Ripley that do not seem to be remotely mentioned in the relied-upon paragraphs. For instance, taking the dependent claims of Claim 1, nothing in paragraphs 21 and 96 of Ripley teaches grouping value correspondences into potential sets such that, for each potential set, at most one value correspondence per target attribute exists. Instead, paragraph 21 summarizes defining match accuracies, and nowhere mentions potential sets of value correspondences much less that, for each potential set, at most one value correspondence per target attribute exists. Paragraph 96 mentions one evident way to define a match accuracy.

Claim 3 recites adding a potential set to a set of candidate sets if only one source relation is used to compute mappings using the potential set, and otherwise adding a potential set to the set of candidate sets only if a join path for the source relations can be identified. The examiner alleges that these are taught in paragraphs 46 and 82 of Ripley, but as usual, his view of what Ripley teaches does not appear to be based on any cognizable word, phrase, or sentence in Ripley. Paragraph 46 is only a brief description of Figure 10. Paragraph 82 merely teaches determining whether a mapping specification exists, and if not one is added to a list, and otherwise an error is returned. No "potential set". No adding a potential set to a set of candidate sets if only one source relation is used to compute mappings using the potential set. No mention of "join path", much less of adding a potential set to the set of candidate sets only if a join path for the source relations can be identified. How the relied-upon sections of Ripley in any rational way relate to the subject matter of Claim 3 is beyond Appellant's understanding.

Since as divulged above Ripley does not have candidate sets, it cannot arrange them into groups as required by Claim 4, much less that each group includes every value correspondence at least once, much less still that the groups establish covers. The relied-upon paragraphs 72-76 seem to be related more to XML than to anything remotely recognizable in Claim 4. Likewise, paragraph 14 of Ripley mentions establishing a hierarchy of datatypes, not groupings of value correspondences that themselves are not datatypes, much less still creating a query for each candidate set in the selected cover, or combining the queries for the cover, as required in Claim 5.

Independent Claim 17 likewise is patentable for reasons stated above. The sections of Ripley being relied on do not teach or suggest generating subsets of value correspondences, much less doing so such that (1) each subset has at most one value correspondence per target attribute, (2) for each subset requiring more

than one source relation to undertake a mapping, a join path can be found between the relations, and (3) each subset includes at least every value correspondence. By way of non-limiting example only, the concept of "join path" is not to be found in Ripley.

Respectfully submitted,

John L. Rogitz
Registration No. 33,549
Attorney of Record
750 B Street, Suite 3120
San Diego, CA 92101
Telephone: (619) 338-8075

JLR:jg

APPENDIX A - APPEALED CLAIMS

1. A computer system, comprising:

a general purpose computer, the computer including logic for undertaking method acts to map data arranged in a source schema into a target schema, the method acts undertaken by the logic including:

receiving at least one value correspondence, each value correspondence representing a function for deriving a value of a target attribute from one or more values of source attributes;

grouping at least some value correspondences into potential sets;

selecting candidate sets from at least some potential sets;

grouping at least some candidate sets into covers; and

using at least one cover, generating at least one query representing a source schema-to-target schema mapping.

2. The computer of Claim 1, wherein the method acts undertaken by the logic to execute the grouping act include:

grouping value correspondences into potential sets such that, for each potential set, at most one value correspondence per target attribute exists.

3. The computer of Claim 2, wherein the method acts undertaken by the logic further include:

adding a potential set to a set of candidate sets if only one source relation is used to compute mappings using the potential set; otherwise

adding a potential set to the set of candidate sets only if a join path for the source relations can be identified.

4. The computer of Claim 3, wherein the method acts undertaken by the logic further include:

arranging candidate sets into groups such that each group includes every value correspondence at least once, the groups establishing covers.

5. The computer of Claim 4, wherein the method acts undertaken by the logic further include:

establishing at least one selected cover;

for each candidate set in the selected cover, creating at least one query; and

combining the queries for the cover.

6. The computer of Claim 5, wherein the method acts undertaken by the logic to establish at least one selected cover include:

ranking the covers by at least one of: a number of candidate sets in each cover, and a number of target attributes; and

presenting ranked covers to a user for selection of one of the covers as the selected cover.

7. The computer of Claim 3, wherein the logic undertakes the act of adding a potential set to the set of candidate sets only if a join path for the source relations can be identified using a spanning tree.

8. The computer of Claim 1, wherein the logic incrementally undertakes the acts of grouping value correspondences into potential sets, selecting candidate sets, grouping candidate sets into covers, and generating queries representing mappings.

9. A computer-implemented method for generating a mapping from a source schema to a target schema, comprising:

generating a mapping based on at least a subset of value correspondences, each value correspondence representing a function for deriving a value of a target attribute from one or more values of source attributes;

allowing a user, in a user interaction, to incrementally add or delete a value correspondence from the subset;

based on the user interaction, generating a new mapping;

presenting a representation of the new mapping to the user such that the user can view the representation; and

permitting the user to add or delete a value correspondence embodied in the new mapping to generate another mapping.

10. The method of Claim 9, wherein the generating act includes:

grouping at least some value correspondences into potential sets;

selecting candidate sets from at least some potential sets;

grouping at least some candidate sets into covers; and

using at least one cover, generating at least one query representing a source schema-to-target schema mapping.

11. The method of Claim 10, further comprising:

grouping value correspondences into potential sets such that, for each potential set, at most one value correspondence per target attribute exists.

12. The method of Claim 11, further comprising:

adding a potential set to a set of candidate sets if only one source relation is used to compute mappings using the potential set; otherwise

adding a potential set to the set of candidate sets only if a join path for the source relations can be identified.

13. The method of Claim 12, further comprising:

arranging candidate sets into groups such that each group includes every value correspondence at least once, the groups establishing covers.

14. The method of Claim 13, further comprising:
establishing at least one selected cover;
for each candidate set in the selected cover, creating at least one query; and
combining the queries for the cover.

15. The method of Claim 14, further comprising:
ranking the covers by at least one of: a number of candidate sets in each cover, and a number of target attributes; and
presenting ranked covers to a user for selection of one of the covers as the selected cover.

16. The method of Claim 12, further comprising adding a potential set to the set of candidate sets only if a join path for the source relations can be identified using a spanning tree.

17. A computer program device comprising:
a computer program storage device readable by a digital processing apparatus; and
a program on the program storage device and including instructions executable by the digital processing apparatus for performing method acts for generating a query representing a source schema-to-target schema mapping, the program comprising:
computer readable code means for establishing plural value correspondences;
computer readable code means for generating subsets of value correspondences such that (1) each subset has at most one value correspondence per target attribute, (2) for each subset requiring more than one source relation to undertake a mapping, a join path can be found between the relations, and (3) each subset includes at least every value correspondence; and
computer readable code means for generating a query using one of the subsets, the query being representative of a source schema-to-target schema mapping.

18. The computer program product of Claim 17, further comprising computer readable code means for sorting the subsets and displaying at least portions of a sorted list of subsets, such that a user can establish a selected subset used to generate the query.

19. The computer program product of Claim 18, wherein the means for generating subsets generates candidate sets, each subset including one or more candidate sets, and the means for sorting sorts the subsets by inverse number of candidate sets.

20. The computer program product of Claim 19, wherein the means for sorting also sorts the subsets by the number of value correspondences in the subsets.

21. The computer program product of Claim 19, wherein the means for generating a query creates at least one query for each candidate set in the selected subset, and then combines the queries for the subset.

22. The computer program product of Claim 21, wherein the means for generating subsets and the means for generating a query are incrementally invoked by a user to selectively add or delete value correspondences from a selected subset.

APPENDIX B - EVIDENCE

Rule 131 Declarations and supporting evidence